instructions contained in each of a plurality of instruction words delimited by instruction delimiting information, said parallel processor comprising:

an instruction issue unit recognizing and, in accordance therewith, selectively issuing each of the basic instructions to a corresponding instruction execution unit to execute the issued basic instruction.

REMARKS

INTRODUCTION:

In accordance with the foregoing, a new title has been added, a new abstract has been added, the specification has been amended, claims 1 and 7 have been amended, and claims 9-12 have been added. No new matter is being presented, and thus approval and entry are respectfully requested.

Claims 1-12 are pending and under consideration. The examiner's rejections are traversed below.

OBJECTION TO THE TITLE:

In the Office Action, at page 2, the title was objected to as not being descriptive. In view of the proposed amended title set forth above, the outstanding objection to the title should be resolved.

CHANGES TO THE SPECIFICATION:

In item 2 on page 2 of the Office Action the Examiner requested the applicants' cooperation in correcting any errors. Changes have been made to the specification only to place it in preferred and better U.S. form for issuance and to resolve the Examiner's objections raised in the Office Action. No new matter has been added.

REJECTION UNDER 35 U.S.C. §102:

In the Office Action, at page 2, claims 1-6 were rejected under 35 U.S.C. §102 in view of Faraboschi et al, U.S. Patent Number 5,930,508 (herein referred to as Faraboschi). The reasons for the rejection are set forth in the Office Action and therefore not repeated. This rejection is traversed and reconsideration is requested.

Faraboschi teaches attaching a dispersal bit to each basic instruction of a compacted instruction set. This allows each instruction to be sent to the corresponding functional unit to execute that basic instruction (Fig. 4 and column 4, line 57 to column 5, line 11). These dispersal bits, which are attached to each basic instruction, occupy the memory along with the delimiter bits which separate the instructions (Figure 5).

This is in contrast with the present invention, which provides the processor with flags, corresponding to respective, individual execution units in the processor, which allow the basic instructions that are concurrently executable to be supplied to the respective execution units according to the flags. These flags are not attached to the basic instructions, as are the dispersal bits in Faraboshci, but are provided to correspond to the respective execution units while being independent of the basic instructions. This allows the instruction codes to be stored in a memory device with only delimiter data, as opposed to each instruction having a dispersal bit attached along with delimiter data. The issue instruction unit of the parallel processor recognizes the instruction codes without the need of dispersal bits.

Also, the method disclosed in Faraboschi increases the hardware size needed for the processor. When each instruction must include a dispersal bit to lead the instruction to its corresponding functional unit, excessive hardware, such as a crossbar switch, must be provided to deliver the instructions according to their dispersal information. This is also in contrast with the present invention, which specifies an execution unit by use of a flag that is provided for the execution unit independently of the basic instructions, thereby providing for a basic instruction to be supplied to the specified execution unit, thus reducing hardware size.

Referring to claim 1 as amended, it is submitted that Faraboschi does not teach a parallel processor, that performs parallel processing of instructions delimited by delimiting information, having an instruction issue unit that "includes an interface having effective bits corresponding to the instruction execution units, the effective bits indicating whether the corresponding instruction execution unit is available." Therefore it is submitted that claim 1

patentably distinguishes over Faraboschi.

Claims 2-6 depend from claim 1 and include all of the features of that claim plus additional features which are not taught or suggested by Faraboschi. Therefore, it is submitted that claims 2-6 also patentably distinguish over Faraboschi.

REJECTION UNDER 35 U.S.C. §103:

In the Office Action, at page 5, claims 7 and 8 are rejected under 35 U.S.C. §103 as being unpatentable over Faraboschi in view of Nair et al, "Exploiting Instruction Level Parallelism in Processors by Caching Scheduled Groups." The reasons for the rejection are set forth in the Office Action and therefore not repeated. The rejection is traversed and reconsideration is requested.

As stated above, Applicant respectfully submits that claim 1 patentably distinguishes over Faraboschi. Since claims 7 and 8 depend from claim 1 and include all of the features of that claim, plus additional features which are not taught or suggested by the prior art, it is submitted that claims 7 and 8 patentably distinguish over the prior art.

CONCLUSION:

In accordance with the foregoing, it is respectfully submitted that all outstanding objections and rejections have been overcome and/or rendered moot. And further, that all pending claims patentably distinguish over the prior art. Thus, there being no further outstanding objections or rejections, the application is submitted as being in condition for allowance which action is earnestly solicited.

If the Examiner has any remaining issues to be addressed, it is believed that prosecution can be expedited by the Examiner contacting the undersigned attorney for a telephone interview to discuss resolution of such issues.

If there are any underpayments or overpayments of fees associated with the filing of this Amendment, please charge and/or credit the same to our Deposit Account No. 19-3935.

Respectfully submitted,

STAAS & HALSEY LLP

Date: June 12, 2003

By: _____
H. J. Staas
Registration No. 22,010

700 Eleventh Street, N.W.
Suite 500
Washington, D.C. 20001
Telephone: (202) 434-1500
Facsimile: (202) 434-1501

CERTIFICATE OF FACSIMILE TRANSMISSION
I hereby certify that this correspondence is being transmitted via facsimile to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450
on June 12, 2003
STAAS & HALSEY
By: _____
Date June 12, 2003

VERSION WITH MARKINGS TO SHOW CHANGES MADE

IN THE SPECIFICATION:

Please AMEND the paragraph beginning at page 2, line 10, as follows:

The instruction execution units EU0 to EUn execute the basic instructions, and notify the instruction issue unit 3 of the end of the execution. The register unit 5 supplies data to the instruction execution units EU0 to EUn, if necessary, and holds the execution results of the instruction execution units EU0 to EUn. The externally connected memory 7 stores [a] an instruction word string to be executed in the parallel processor 10. The memory 7 also stores necessary data for the execution units EU0 to EUn to execute instructions, and data as the execution results.

Please AMEND the paragraph beginning at page 2, line 22, as follows:

FIG. 2 shows the formats of instruction words to be supplied to a parallel processor having four instruction execution units EU0 to EU3. As shown in FIG. 2, each instruction word is made up of a basic instruction EI and a do-nothing instruction NOP. If the number of basic instructions contained in one instruction word to be executed in parallel is smaller than the number of the instruction execution units EU0 to EU3, the proportion of do-[noting]nothing instructions is large.

Please AMEND the paragraph beginning at page 3, line 10, as follows:

With the super scalar technique, there is also a problem that a large-scale circuit is needed to increase the number of [instructoins] instructions to be executed in parallel.

Please AMEND the paragraph beginning at page 9, line 19, as follows:

FIG. 5 shows the formats of instruction words to be supplied to the parallel processors of the first embodiment. Each instruction word is made up of one or more basic instructions EI and at least one of instruction word delimiting fields 0 and 1. The basic instruction word length is either 1 or 2. The upper row of FIG. 5 indicates an instruction word having a basic instruction

word length of 2, consisting of a basic instruction word made up of an instruction word delimiting field 0 and a basic instruction EI, and another basic instruction word made up of an instruction word delimiting [filed] field 1 and a basic instruction EI. The lower row of FIG. 5 indicates an instruction word having a basic instruction word length of 1, consisting of an instruction word delimiting field 1 and a basic instruction EI.

Please AMEND the paragraph beginning at page 10, line 20, as follows:

Based on the instruction word delimiting fields 0 and 1 contained in the instruction words supplied from the cutting unit 316, the instruction issue unit 72 recognizes each basic instruction EI, and issues each basic instruction EI selectively to one of the instruction execution units EU0 and EU1 via the selectors 355 and 356. Accordingly, if a basic instruction EI following an instruction word delimiting field 0 is issued to the instruction execution unit EU0, [while] a basic instruction EI following an instruction word delimiting field 1 is issued to the instruction execution unit EU1. The selectors 355 and 356 are controlled by the control unit 370. When the execution of one instruction word is completed, the corresponding basic instruction EI is supplied to the instruction execution units EU0 and EU1 via the selectors 355 and 356.

Please AMEND the paragraph beginning at page 14, line 5, as follows:

As described so far, the parallel processor 21 of this example can have the same effects as the parallel processor 20 of Example 1, and efficiently and accurately performs the parallel processing of the basic instructions EI. Thus, more reliable operations can be achieved.

Please AMEND the paragraph beginning at page 16, line 26, as follows:

For simplification of the drawing, only two instruction passages from an instruction register 348 to the two instruction execution units LU0 and [LU1] IUO are shown in FIG. 8. However, it should be understood that there are the other instruction passages to the instruction execution units IU1, FU0, FU1, and BU0, as shown in FIG. 7.

Please AMEND the paragraph beginning at page 18, line 22, as follows:

FIG. 10 is a circuit diagram of the conversion unit 115 shown in FIG. 8. As shown in

FIG. 10, the conversion unit 115 comprises transmission lines L1 and L2, BI detectors BD1 and BD2, FI detectors FD1 and FD2, II detectors ID1 and ID2, LI detectors LD1 and LD2, buffers 155 to 158, AND gates 163 to 166, 185, and 186, exclusive OR gates 187 to 190, selectors 209 to 212, and OR gates 199 to 202.

Please AMEND the paragraph beginning at page 19, line 7, as follows:

The FI detector FD1 is connected to the transmission line L1, and the FI detector FD2 is connected to the transmission line L2. The buffer 156 is connected to the FI detector FD1, and the AND gate 164 is connected to the FI detectors FD1 and FD2. The two input terminals of the exclusive OR gate 187 are connected to the input node and the output node, respectively, of the buffer 156. The two input terminals of the exclusive [IR] OR gate 188 are connected to the output node of the AND gate 164 and the FI detector FD2, respectively. The AND gate 185 is connected to the two exclusive OR gates 187 and 188. The selector 210 is connected to the transmission lines L1 and L2, the buffer 156, and the AND gate 164. The OR gate 200 is connected to the buffer 156 and the AND gate 164.

Please AMEND the paragraph beginning at page 21, line 24, as follows:

As the second basic instruction FI is detected, the BI detector BD2, the II detector ID2, and the LI detector LD2 output non-detection signals of logic 0. Accordingly, the selectors 209, 211, and 212 do not select the second basic instruction transmitted through the transmission line L2. Since neither first nor second basic instructions to be [executed] execution by the instruction executed units LU0, IU0, IU1, and FU1 are detected, the effective [bet] bit V of logic 0 is outputted from each of the OR gates 201 and 202, and the AND gates 185 and 186.

Please AMEND the paragraph beginning at page 22, line 26, as follows:

The conversion unit 115 having the above structure operates in the same manner as the conversion unit 115 shown in FIG. 10. In the following, an operation of the conversion unit 115 in a case where an instruction word made up of basic instructions BI, FI, FI, and II is supplied to the conversion unit 115 will be described. First, the first basic instruction BI is transmitted through the transmission line L1. The BI detector BD1 then detects the basic instruction BI and

supplies a detection signal of logic 1 to the buffer 159. At this point, each of the AND gates 167 to 169 outputs a logic 0 signal. In accordance with the detection signal supplied from the buffer 159, the selector 213 selects the first basic instruction BI and outputs the first basic instruction BI, that is an instruction to be executed by the instruction execution unit BU0, to the instruction issue unit 74. At the same time as the output of the first basic instruction BI, the OR gate[s] 203 outputs the effective bit V of logic 1 in accordance with the detection signal supplied from the buffer 159. As the first basic instruction BI is detected, the FI detector FD1, the II detector ID1, and the LI detector LD1 output non-detection signal of logic 0. Accordingly, the selectors 214, 216, and 218 do not select the first basic instruction BI transmitted through the transmission line L1.

Please AMEND the paragraph beginning at page 25, line 15, as follows:

In this example, the instruction fetch unit 48 can also fetch an instruction word containing basic instructions that have already been arranged in accordance with the arrangement of the instruction execution units in advance. In such a case, the basic instructions are arranged in advance so that the circuit size required for rearranging the basic instructions in the instruction fetch unit 48 can be reduced.

Please AMEND the paragraph beginning at page 30, line 12, as follows:

For simplification of the drawing, only the instruction passages from an instruction register 351 to the two instruction execution units LU0 and IU0 are shown, and the other instruction passages to the instruction execution units IU1, FU0, FU1, and BU0 are omitted in FIG. 17. Likewise, only the two execution complete signals LUc and IUc0 are shown as signals to be supplied to the judgment unit 104, [but] and the other execution complete signals are omitted in FIG. 17.

Please AMEND the paragraph beginning at page 33, line 8, as follows:

FIG. 21 shows the structures of the instruction fetch unit 53 and the instruction issue unit 79 of the parallel processor 27 shown in FIG. 20. The instruction fetch unit 53 and the instruction issue unit 79 have the same structures as the instruction fetch unit 50 and the instruction issue unit 76 shown in FIG. 15, except that the instruction issue unit 79 further includes the judgment unit 106 connected between an instruction register 353 and a control unit 376. Based on a supplied basic instruction, the judgment unit 106 determines whether or not a

basic instruction to be issued has the data dependency or control dependency, or causes resource sharing. The judgment results are[a] reported to the control unit 376. If the basic instruction to be issued has the data dependency or control dependency, or causes resource sharing, the instruction issue unit 79 issues the basic instruction only after the execution complete signals LUc and IUc0 are supplied.

Please AMEND the paragraph beginning at page 34, line 29, as follows:

As shown in FIGS. 22 to 27, parallel processors 28 to 33 in accordance with a third embodiment of the present invention each comprises an instruction fetch unit 54-59 connected to the memory 12, an instruction issue unit 80-85 connected to the instruction fetch unit 54-59, instruction execution units LU0, IU0, IU1, FU0, FU1, MU0, MU1, and BU0, and a register unit 100 connected to all the instruction execution units. Here, the instruction execution units MU0 and MU1 are special-purpose arithmetic instruction execution units that execute special-purpose arithmetic instructions. When the execution of special-purpose arithmetic instructions is completed, the instruction execution units MU0 and MU1 notify the instruction issue unit 80-85 of the [complete] end of the execution.

Please AMEND the paragraph beginning at page 35, line 8, as follows:

In the following, the parallel processors in accordance with the third embodiment of the present invention will be described by way of a case where the maximum basic instruction word length contained in one instruction word is 2. It should be understood that the same effects can be obtained in a case where the maximum instruction word length contained in one instruction word is 3 or more [greater].

IN THE CLAIMS:

Please AMEND claims 1 and 7. Please ADD claims 9-12.

The remaining claims are reprinted, as a convenience to the Examiner, as they presently stand before the U.S. Patent and Trademark Office.

1. (ONCE AMENDED) A parallel processor [that performs]performing parallel

processing of one or more basic instructions contained in each of a plurality of instruction words delimited by instruction delimiting information, said parallel processor comprising:

a plurality of instruction execution units [that perform]performing processes in accordance with corresponding, [to] supplied basic instructions in parallel;

an instruction fetch unit [that fetches]fetching the instruction words one by one in accordance with the instruction delimiting information; and

an instruction issue unit [that] recognizing and, in accordance therewith, selectively [issues]issuing each of the basic instructions supplied from the instruction fetch unit to one of the corresponding instruction execution units to execute [an]the issued basic instruction.

2. (NOT AMENDED) The parallel processor as claimed in claim 1, wherein the plurality of instruction execution units all have the same structure.

3. (NOT AMENDED) The parallel processor as claimed in claim 1, wherein:

at least two of the instruction execution units have different structures from each other; and

the instruction fetch unit rearranges the basic instructions contained in each of the fetched instruction words, in accordance with arrangement of the plurality of instruction execution units, and then supplies the rearranged basic instructions to the instruction issue unit.

4. (NOT AMENDED) The parallel processor as claimed in claim 1, wherein:

at least two of the instruction execution units have different structures from each other; and

the instruction issue unit rearranges the basic instructions contained in each of the instruction words supplied from the instruction fetch unit, in accordance with arrangement of the plurality of instruction execution units, and then supplies the rearranged basic instructions to the instruction execution units.

5. (NOT AMENDED) The parallel processor as claimed in claim 1, wherein:

at least two of the instruction execution units have different structures from each other;

the instruction fetch unit rearranges the basic instructions contained in each of

the fetched instruction words, in accordance with arrangement of the instruction execution units, and then supplies the rearranged basic instructions to the instruction issue unit; and

the instruction issue unit further rearranges the basic instructions contained in each of the instruction word supplied from the instruction fetch unit, in accordance with the arrangement of the instruction execution units, and then supplies the rearranged basic instructions to the instruction execution units.

6. (NOT AMENDED) The parallel processor as claimed in claim 3, wherein:

at least two of the instruction execution units have different structures from each other; and

the instruction fetch unit fetches an instruction word that contains basic instruction arranged in advance in accordance with the arrangement of the instruction execution units.

7. (ONCE AMENDED) The parallel processor as claimed in claim 1, wherein, depending on the type of a basic instruction being currently executed by one of the instruction execution units, the instruction issue unit issues a next basic instruction before the execution of [a]the basic instruction being currently executed is completed.

8. (NOT AMENDED) The parallel processor as claimed in claim 7, wherein, if a supplied basic instruction does not have data dependency or control dependency, or does not share resources with a basic instruction being currently executed by one of the instruction execution units, the instruction issue unit issues the supplied basic instruction before the execution of the basic instruction being currently executed is completed.

9. (NEW) The parallel processor as claimed in claim 1, wherein the instruction issue unit further comprises an interface corresponding to the instruction execution units indicating whether the corresponding instruction execution unit is available.

10. (NEW) A parallel processor as claimed in claim 9, wherein the instruction issue unit further comprises a table for setting effective bits to indicate availability of the corresponding

instruction execution unit.

11. (NEW) A parallel processor as claimed in claim 10, wherein a first instruction word format is converted into a second instruction word format by the table, the first instruction word format indicating an arrangement of instruction words from the instruction fetch unit, and the second instruction word format indicating an arrangement of instruction words which corresponds to the instruction execution units.

12. (NEW) A parallel processor as claimed in claim 9, wherein the instruction issue unit further comprises a conversion unit for converting a first instruction word format into a second instruction word format on the basis of effective bits, corresponding to the instruction execution units, indicating whether the corresponding instruction execution unit is available.

13. (NEW) A parallel processor as claimed in claim 12, wherein the first instruction word format indicates an arrangement of instruction words from the instruction fetch unit, and the second instruction word format indicates an arrangement of instruction words which corresponds to the instruction execution units.

14. (NEW) A parallel processor performing parallel processing of one or more basic instructions contained in each of a plurality of instruction words delimited by instruction delimiting information, said parallel processor comprising:

an instruction issue unit recognizing and, in accordance therewith, selectively issuing each of the basic instructions to a corresponding instruction execution unit to execute the issued basic instruction.

IN THE ABSTRACT:

Please AMEND the abstract as follows:

ABSTRACT

A parallel processor [that]performs efficient parallel processing [is provided. The parallel processor, which performs parallel processing]of one or more basic instructions contained in each of a plurality of instruction words delimited by instruction delimiting information.[,] The processor includes: a plurality of instruction execution units [that]performing

processes in accordance with corresponding, [to]supplied basic instructions in parallel; an instruction fetch unit [that fetches]fetching the instruction words one by one in accordance with the instruction delimiting information; and an instruction issue unit [that issues]recognizing and, in accordance therewith, selecting each of the basic instructions contained in each of the instruction words fetched by the instruction fetch unit to a corresponding [one of the]instruction execution unit[s] to execute the basic instruction.

FIG.4

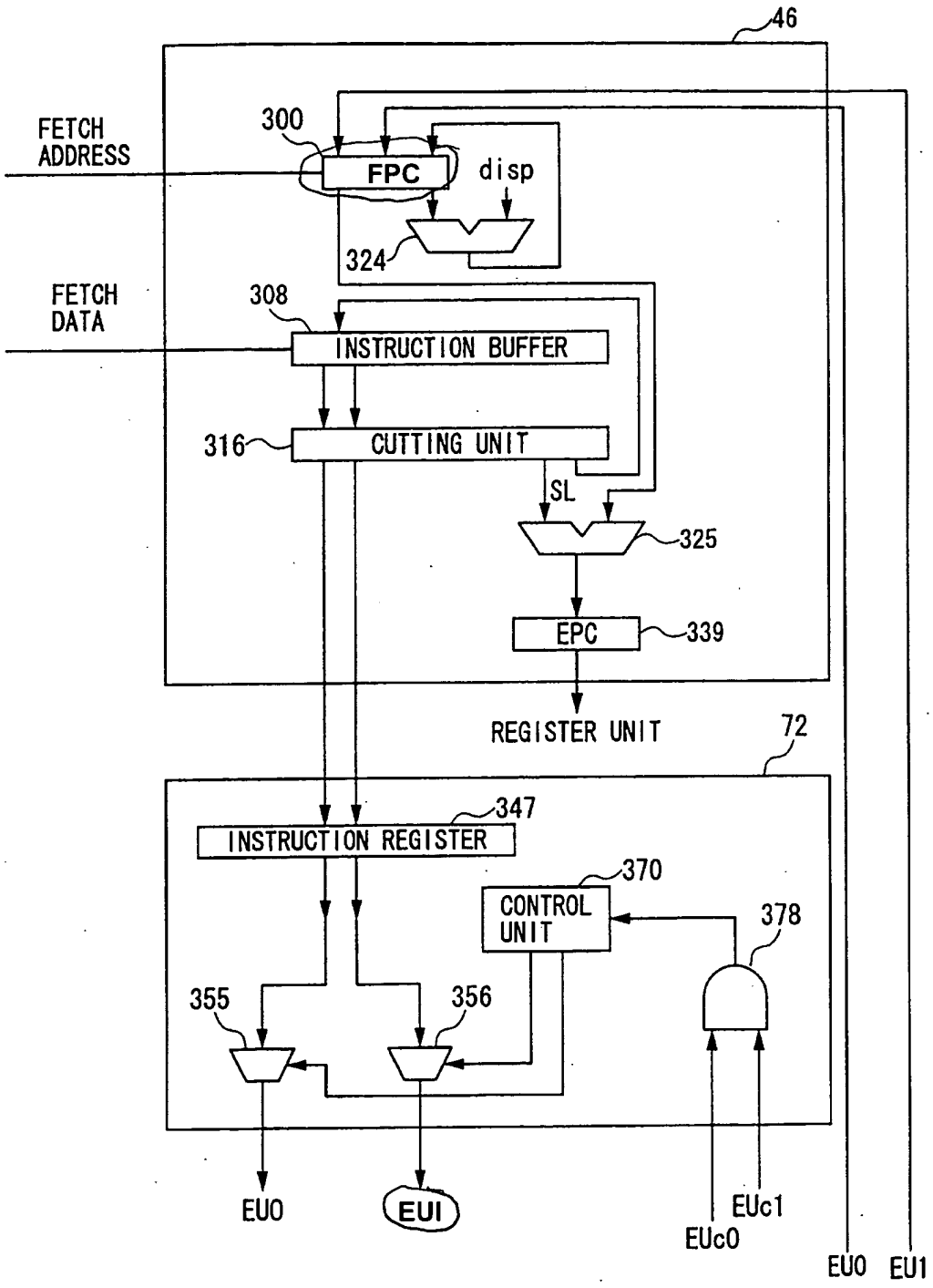

FIG.8

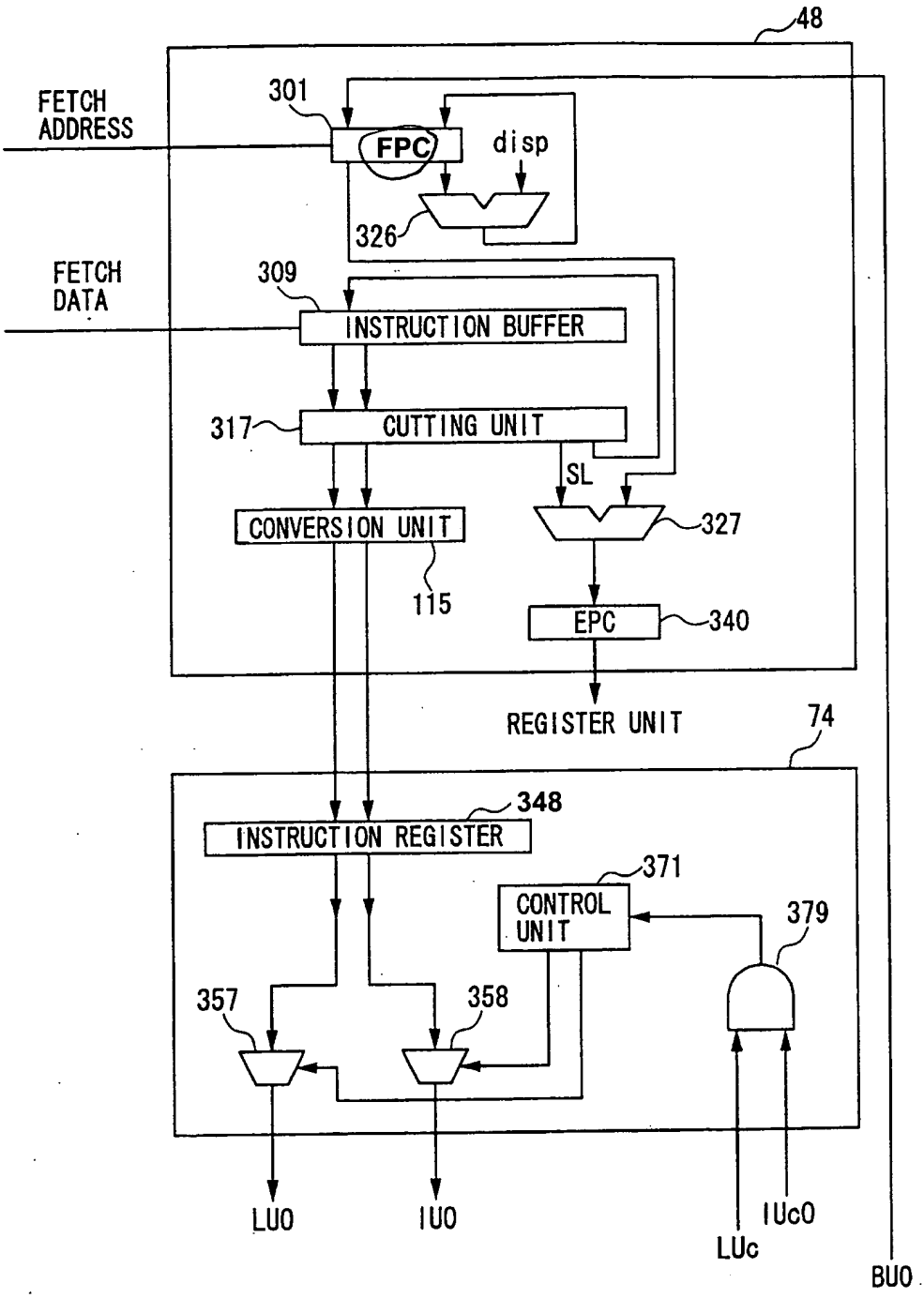

FIG.11